

University of Groningen

Device-based image matching with similarity learning by convolutional neural networks that exploit the underlying camera sensor pattern noise

Bennabhaktula, Swaroop; Alegre, Enrique; Karastoyanova, Dimka; Azzopardi, George

DOI:

[10.5220/0009155505780584](https://doi.org/10.5220/0009155505780584)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2020

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Bennabhaktula, S., Alegre, E., Karastoyanova, D., & Azzopardi, G. (2020). *Device-based image matching with similarity learning by convolutional neural networks that exploit the underlying camera sensor pattern noise*. 578-584. <https://doi.org/10.5220/0009155505780584>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).





The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Device-based Image Matching with Similarity Learning by Convolutional Neural Networks that Exploit the Underlying Camera Sensor Pattern Noise

Guru Swaroop Bennabhaktula¹^a, Enrique Alegre²^b, Dimka Karastoyanova¹^c
and George Azzopardi¹^d

¹*Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence,
University of Groningen, The Netherlands*

²*Group for Vision and Intelligent Systems, Universidad de León, Spain*

Keywords: Source Camera Identification, Image Forensics, Sensor Pattern Noise.

Abstract: One of the challenging problems in digital image forensics is the capability to identify images that are captured by the same camera device. This knowledge can help forensic experts in gathering intelligence about suspects by analyzing digital images. In this paper, we propose a two-part network to quantify the likelihood that a given pair of images have the same source camera, and we evaluated it on the benchmark Dresden data set containing 1851 images from 31 different cameras. To the best of our knowledge, we are the first ones addressing the challenge of device-based image matching. Though the proposed approach is not yet forensics ready, our experiments show that this direction is worth pursuing, achieving at this moment 85 percent accuracy. This ongoing work is part of the EU-funded project 4NSEEK concerned with forensics against child sexual abuse.


1 INTRODUCTION


With the rapid adoption and consumption of digital content, there have been many instances of illicit material of children being circulated on the Internet, especially in the darknet. Today, law enforcement agencies (LEAs) require forensic tools which can help them to investigate more effectively and efficiently such digital content. The EU-funded 4NSEEK project¹, to which this work belongs, is aimed to develop a forensic tool by various partners in the industry and academia with the cooperation of police agencies in the European Union. The project is focused on fighting against child sexual abuse and the distribution of its contents across the internet. One desired functionality is device-based image matching, that is the determination whether any two or more seized images were captured by the same camera device. Here we report the ongoing work in this direction.


Just as the bullet traces in a crime scene become


a piece of evidence for a weapon, a digital image can become an evidence for a camera. This is possible when we can extract fingerprints from images that (uniquely) characterize the source camera device. Extraction and identification of these fingerprints become more challenging when the photographs are subject to compression, post-processing, and computational photography, among others. Every processing step that alters the original RAW image, including the operations that are performed on the captured image within the camera, plays a role in altering the fingerprint. Together with the increasing use of image processing tools, the extraction of fingerprints becomes even more challenging.

The camera signature is embedded in the captured image in the form of noise and some artefacts. Our goal is to extract these fingerprints from given images and use them to determine whether the concerned images were captured by the same camera device. We would like to bring out a subtle difference between the terms camera model and camera device, with the former referring the type of camera (e.g. Nikon D200) and the latter refers to a specific manufactured device (e.g. Nikon D200 - 1, where the last digit represents the unique identifier for the manufactured

^a <https://orcid.org/0000-0002-8434-9271>

^b <https://orcid.org/0000-0003-2081-774X>

^c <https://orcid.org/0000-0002-8827-2590>

^d <https://orcid.org/0000-0001-6552-2596>

¹<https://www.incibe.es/en/european-projects/4nseek>

Nikon D200 devices). In this work, we address image matching by using signatures of the source camera devices.

More formally, the problem that we address in this work is the following: given a pair of images, how likely are they were both captured using the same camera device. We restrict our analysis and discussions to the publicly available Dresden (Gloe and Böhme, 2010) image data set. We propose a convolutional neural network (CNN) based architecture which is in line with the design of the CNN proposed by Mayer and Stamm (2018) for camera model identification.

The rest of the paper is organized as follows. We start by presenting an overview of the traditional and state-of-the-art approaches in Section 2. In Section 3, we describe the approach for feature extraction and classification of the proposed source camera identification. Experimental results along with the dataset description are provided in Section 4. We provide a discussion of certain aspects of the proposed work in Section 5 and finally, we draw conclusions in Section 6.

2 RELATED WORK

The camera signature is embedded in the captured image in the form of noise and some artefacts. In Figure 1 we illustrate a hierarchical representation of noise classification which we adopt from Lukáš et al. (2006). Even when the camera sensor is exposed to a uniformly lit scene the resulting image pixels are not uniform. This non-uniformity is caused due to shot noise and pattern noise. *Shot noise* is a temporal random noise and varies from frame to frame. This component of noise can be suppressed to a large extent by frame averaging. *Pattern noise* is defined as any noise component that survives frame averaging (Holst, 1998). This stability and uniqueness over time makes pattern noise a candidate for camera signature.

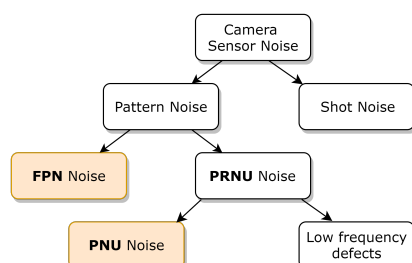


Figure 1: Topology of digital camera sensor noise. Note that only FPN (Fixed Pattern Noise) and PNU (Pixel Non-uniformity Noise), which are highlighted in yellow, contain the fingerprint that can be used to uniquely identify a sensor.

The two main components of pattern noise are FPN (fixed pattern noise) and PRNU (photo response non-uniform noise). The FPN is an additive noise which is a consequence of dark currents (Holst, 1998). Dark currents are responsible for pixel-to-pixel differences when the sensor is not exposed to any light. Some modern digital cameras do offer long exposure noise reduction that automatically subtracts a dark frame from the captured image. This helps in removing the FPN artefacts from the captured image. This is, however, not a de-facto standard and is not implemented by all consumer camera manufacturers.

PRNU is further classified into PNU (pixel non-uniformity noise) and noise caused by low frequency defects. PNU noise is mainly caused due to imperfections and defects introduced into the sensor during the semiconductor wafer fabrication process. This inhomogeneity results in different sensitivity of pixels to light. The nature of PNU is such that even the sensors that are fabricated from the same wafer exhibit different PNU patterns. As mentioned by Lukáš et al. (2006), light refraction on dust particles, optical surfaces, and zoom settings also contribute to PRNU noise. These *low frequency* components are not characteristic of the sensor, hence they should be discarded when capturing the noise profile for a sensor from its image.

2.1 Traditional Approaches

To the best of our knowledge, one of the earliest published works in camera detection was done by Geradts et al. (2001). The authors showed that every CCD (charge coupled device) sensor exhibits few random pixels which are defective. These pixels can be identified under controlled temperatures. Repeated experiments showed that the location of such defective pixels always remain the same. The authors built a probabilistic model based on the location of defective pixels. The detection of such pixels is then left to visual inspection. Kharrazi et al. (2004) proposed 34 handcrafted features combined with an SVM classifier (Chang and Lin, 2011) to distinguish between images taken by Nikon E-2100, Sony DSC-P51, and Canon (S100, S110, S200) cameras. The authors extracted these features from both spatial and wavelet domains and carried out their experiments on a proprietary data set.

Kurosawa et al. (1999) were the first to consider FPN for source sensor identification. They established that this type of noise exhibits itself in images and is unique for each camera. The authors observed that the power of FPN is much less than the random noise. Hence, in order to suppress random noise and

highlight FPN, they averaged 100 dark frames, which were captured by covering the camera lens. They performed experiments on nine different cameras, eight cameras of which exhibited FPN while the CCD-TRV90 Sony camera did not. Lukáš et al. (2006) have extended on this work by factoring in PRNU noise in addition to the FPN. For each camera under investigation the authors generated a reference pattern noise, which serves as a unique identification fingerprint for the camera. The reference pattern is generated by averaging the noise obtained from multiple images using a denoising filter. The novelty of that approach is the generation of a camera signature without having access to the camera. Finally, the correlation was used to establish the similarity between the query and reference patterns.

Li (2010) studied the noise patterns and observed that the scene details have stronger signal components while the true camera noise has weaker signals. Hence, the stronger noise signal components in the residual image should be less trustworthy. Based on this observation, an enhanced noise fingerprint is extracted by assigning less significant weights to strong components of the noise signal.

A variety of techniques have been proposed which account for CFA (color filter arrays) demosaicing artefacts. These methods identify the source camera of an image based on the traces left behind by the proprietary interpolation algorithm used for each digital camera. Notable among these works include those by Bayram et al. (2005); Swaminathan et al. (2007), and more recent one by Chen and Stamm (2015).

2.2 Approaches based on Deep Learning

In the last few years, deep learning based approaches have also been applied in the field of image forensics. Several CNN-based systems have been proposed to detect traces of image inpainting (Zhu et al., 2018), effects of image resizing and compression (Bayar and Stamm, 2017), and median filtering detection (Chen et al., 2015), among other image forensic tasks.

Researchers have additionally proposed to apply CNNs for the identification of source cameras of given images (Tuama et al., 2016; Bondi et al., 2016). Most of the deep learning algorithms follow an approach of extracting noise patterns by suppressing the scene content. Interestingly, the first deep learning architectures for image denoising are inspired by the work in steganalysis (Qian et al., 2015). This is a technique that adds a first layer with a high pass filter which could either be fixed or trainable (Bayar and Stamm, 2016). Zhang et al. (2017) were the first ones

to successfully do residual learning by a deep architecture. Residual learning is useful for camera sensor identification because the camera signature is often embedded in the residual images, which are obtained by subtracting the scene content from an image. The authors, proposed a deep CNN model that was able to handle unknown levels of additive white Gaussian noise (AWGN). The CNN model was effective in several image denoising tasks, as opposed to traditional model-based designs (Kharrazi et al., 2004), which focused on detecting specific forensic traces.

The drawbacks of many of the proposed approaches are that they target specific types of forensic traces. For example, researchers have proposed methods that exclusively target CFA interpolation artefacts, chromatic aberration, assume a fixed level of Gaussian noise, and more. This is not an ideal assumption when developing real world applications for forensic investigators. Here, we work with an open set of forensic traces.

The works which are very close to the ideas we propose are those by Cozzolino and Verdoliva (2019) and Mayer and Stamm (2018). Both approaches follow an open set of camera models. Many approaches that rely only on a closed set of camera models rely on prior knowledge from the source camera models. It looks almost impossible to use all existing camera models for training such models, and moreover, the scalability of such systems could be a challenge.

Cozzolino and Verdoliva (2019) designed a CNN which extracts a camera model fingerprint (as an image residue) known as the *noiseprint*. The authors use the CNN architecture proposed by Qian et al. (2015) and trained it in a Siamese configuration to highlight the camera-model artefacts. Their work primarily focused on the extraction of noiseprint for camera models and on detecting image forgeries.

The CNN architecture that we adopt in our work is inspired by the work of Mayer and Stamm (2019). The authors have proposed a system called forensic similarity which determines if two image patches contain the same forensic traces or not. They proposed a two-part network. The first one is a feature extractor and the second part is a similarity network, which determines if two features come from the same source camera model. Patch-based systems do not account for the spatial locality. Therefore, instead of relying only on the patches our proposed system takes the whole image for feature extraction. By considering the whole image the network has the possibility to learn the spatial locality in addition to the sensor pattern noise.

3 PROPOSED APPROACH

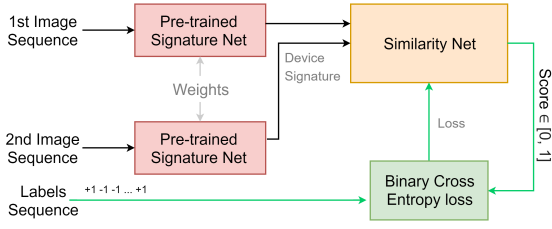


Figure 2: Proposed workflow.

The proposed method compares two input images and generates a score indicating the similarity between the source camera devices that took the concerned images. In Figure 2 we depict the high level workflow of the proposed method. The approach is divided into two phases. In the first phase, we train a CNN called henceforth as *signature network*, responsible for extracting the camera signature from an image. The second stage involves computing the similarity between two image signatures. The similarity function is formulated by training a neural network, which we call *similarity network*.

A two-phase learning approach gives us the ability to independently fine tune signature extraction and similarity comparison. The training of the networks does not need the availability of ground truth noise residuals. It, therefore, allows us to have a more practical approach, as forensic investigators will not have access to the noise residuals for learning the camera signatures.

3.1 Learning Phase I

The first phase in this approach begins with the training of a signature network, which is defined as follows.

Let the space of all RGB images be denoted by \mathbb{I} . The signature network is trained on a subset of images from \mathbb{I} . The trained network is then truncated at a features extraction layer (Layer # 5, labeled Dense signature in Table 1), which we denote by f_{sig} . It is a feed-forward neural network function $f_{sig} : \mathbb{I} \times \mathbb{I} \rightarrow \mathbb{S}$, where \mathbb{S} is a space of all signatures. We define the signature extraction operation, as follows:

$$S = f_{sig}(I) \quad (1)$$

$\forall I \in \mathbb{I}$, where $S \in \mathbb{S}$.

The signature network consists of four convolutional layers followed by two fully connected layers. A summary of all these layers is shown in Table 1. Note that the number of devices in the final fully connected layer represents the number of camera models present in the training set. The variable f_{sig} represents

Table 1: The proposed CNN architecture of the signature network. It consisting of 4 blocks of convolutional layers and 2 blocks of fully connected dense layers. The highlighted row indicates the layer at which we truncate the network and use the resulting 1024-element feature vector as signature.

#	Layers	Activation	Dims	Repeat
1	Conv 2d Batchnorm Max pool	– tanh –	$96 \times 7 \times 7$ – 3×3	$\times 1$
2,3	Conv 2d Batchnorm Max pool	– tanh –	$64 \times 5 \times 5$ – 3×3	$\times 2$
4	Conv 2d Batchnorm Max pool	– tanh –	$128 \times 1 \times 1$ – 3×3	$\times 1$
5	Signature	tanh	1024	$\times 1$
6	Dense Dense	tanh softmax	200 # devices	$\times 1$

the trained network truncated at block 5 (see Table 1). This gives us a signature of 1024 elements in size.

3.2 Learning Phase II

The goal of the second phase is to map the signatures of pairs of images to a similarity score that gives an indication of whether the input pair comes from the same or different source. To this extent, we train a neural network in a Siamese fashion that determines the similarity between a pair of signatures extracted using the signature network. Let S_1 and S_2 be two signatures extracted from the signature network; $S_1 = f_{sim}(I_1)$ and $S_2 = f_{sim}(I_2)$. The labeled data for training the similarity network is then generated according to the following condition:

$$S_{label}(S_1, S_2) = \begin{cases} 1, & \text{If } I_1 \text{ and } I_2 \text{ come from} \\ & \text{the same source camera} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

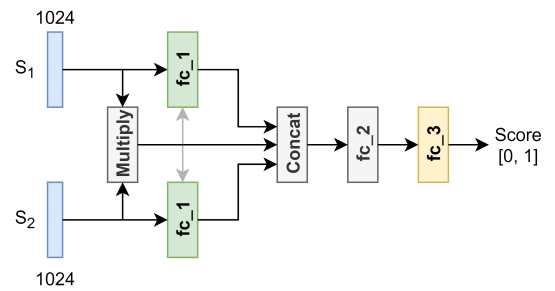


Figure 3: The proposed neural network architecture of the Similarity Network.

The similarity network learns the mapping $f_{sim} : \mathbb{S} \times \mathbb{S} \rightarrow [0, 1]$, and its architecture is depicted in Figure 3. The first layer is a fully connected dense layer

fc_1 containing 2048 neurons with ReLU activation, which takes as input the signatures S_1 and S_2 of a given pair of images. Then, we combine the outputs from the first dense layer along with an element wise multiplication of S_1 and S_2 into a single vector and feed it to fc_2 , which is a dense fully connected layer with ReLU activations. This is finally connected to a single neuron with a sigmoid activation. Once the similarity network is trained, we can use both networks together in a pipeline to determine the similarity for any given pair of input images.

$$score = f_{sim}(f_{sig}(I_1), f_{sig}(I_2)) \quad (3)$$

We experimentally determine a threshold η for the score given by the network. The pairs of images whose similarity score is above η are classified as similar, otherwise as different.

4 PRELIMINARY EXPERIMENTS AND RESULTS

4.1 Dataset

We used the publicly available Dresden dataset (Gloe and Böhme, 2010) in our experiments for image matching based on source camera identification. It consists of images from various indoor and outdoor scenes acquired under controlled conditions.

Many camera model identification approaches have been presented but due to a lack of benchmark datasets, it is often hard to directly compare the performance of different methods. The Dresden dataset was made available in 2010 and since then it has seen widespread use in image forensics that also go beyond source camera identification. The Dresden dataset comes with three subsets of data, one of which is called JPEG, which was intended for the study of model specific JPEG compression algorithms. The JPEG set consists of 1851 images taken by 34 different camera devices that belong to 25 camera models. We discard the three camera devices (FujiFilm_FinePixJ50_0, Ricoh_GX100_3, Sony_DSC-T77_1) that contain only one image each and work with the remaining 31 devices. Though the content is limited to two indoor scenes, it is of interest to understand the source camera device identification in the presence of JPEG compression artefacts. The other two subsets, which consist of dark frames and natural images, were not considered in our study.

4.2 Experiments

In a random stratified manner, we used 70% of the data for training and validation. The remaining 30% of data was left for our tests.

In the first training phase the signature network was trained for 15 epochs, with categorical cross entropy as the loss function and stochastic gradient descent (SGD) as the optimizer. For the optimization task, we set the learning rate to 0.001, the momentum to 0.95, and the decay to 0.0005. Convergence was reached after the 5th epoch where the validation loss started to fluctuate while the training loss remained roughly the same, and we fixed the network with weights obtained at the end of the 5th epoch. The training was done on an NVIDIA RTX 2070 GPU. All the extracted signatures were stored in a database, which provided easy access in the second half of the experiments.

In the second phase, where we train the similarity network, we generated labeled pairs of signatures according to Equation 2. All the 1294 (70% of the full data set of 1851 images) training and validation images used for learning the signature network generated $^{1294}C_2$ pairs of labeled signatures data. The similarity network was trained in a Siamese fashion using binary cross entropy as the loss function along with an SGD optimizer. The network was trained for 30 epochs, with a learning rate of 0.005 and a decay factor of 0.5 for every 3 epochs.

For the systematic evaluation of the trained network, we performed a series of experiments. A single experiment involves choosing a pair of camera devices and generating 100 random pairs of images with replacement. Each pair consists of an image from each of the two concerned camera devices. The trained network was used to predict the similarity score for each of the pairs. The similarity score is converted to 1 (similar) or 0 (not similar) based on a threshold which we determined from the evaluation on the validation set. We set the threshold to 0.99 as it provided the maximum F1-score on the validation set. We normalized the resulting 100 scores by averaging them in order to get a value between 0 and 1 for the comparison of images coming from two camera devices.

For evaluation of the network, we considered all possible pairs of the 31 camera devices resulting in a 31×31 experiments. We used Algorithm 1 below to generate a similarity matrix of 31×31 elements, where each element is the normalized similarity score of the corresponding camera devices. Figure 4 shows the resulting similarity matrix of our test data. The overall accuracy is 85%.

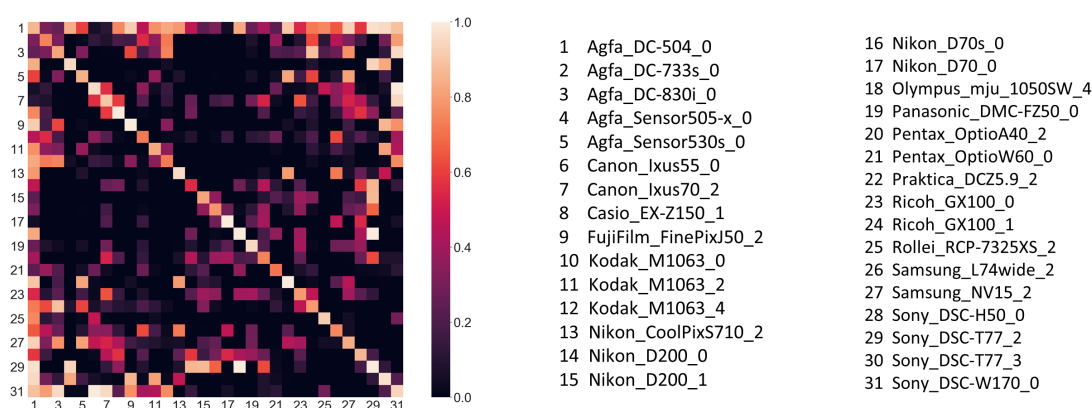


Figure 4: Similarity matrix for the 31 camera devices in the test set. A score closer to 1 indicates a high similarity between the images taken from the corresponding pairs of cameras. Similarity values along the diagonal correspond to the similarity between images taken from the same cameras. Ideally, the similarity matrix has ones along diagonal, and zeros elsewhere.

Algorithm 1: Similarity matrix computation.

```

procedure SIMILARITY MATRIX
   $C \leftarrow \{C_1, C_2, \dots, C_N\}$   $\triangleright N$  cameras
  for  $i \leftarrow 1 : N$  and  $j \leftarrow 1 : N$  do
    Randomly sample 100 image pairs
    from the subspace  $C_i \times C_j$ 

    Predict the Source Similarity for
    the concerned 100 pairs of images
    using Equation 3.

    Compute the accuracy
  end for
  return accuracy for  $N \times N$  experiments
end procedure

```

5 DISCUSSION AND FUTURE WORK

As can be seen from Figure 4, in general, the model is able to detect images coming from the same camera devices. There are, however, some instances where the network gets confused with images coming from the same camera model. This can be seen with camera models (Ricoh_GX100_0, Ricoh_GX100_1), (Nikon_D70_0, Nikon_D70s_0). This could be because the same camera models are subject to the same manufacturing process. Thereby, resulting in similar imperfections or artefacts. We need to first investigate the noise differences between the same camera models, before trying to investigate the noise patterns together from all the devices. This approach might give us a better insight into the challenges between the same camera models.

It is also evident that the devices from the brands

Agfa and Sony get confused with several other camera devices in our evaluation. We suspect this is due to the presence of a large number of images in the data set coming from Agfa (around 25 percent), which may have caused some bias in the learned networks. We will address this problem by investigating different approaches that deal with unbalanced training sets.

The approach that we propose mimics the practical situation faced by forensic experts, where they only have a collection of images without knowing their actual source. Among others, investigators are interested to determine whether two or more images were taken by the same camera, irrespective of what camera it is. That information can help them identifying the offender or to compile stronger evidence. To the best of our knowledge, this is the first attempt that addresses the problem of device-based image matching.

6 CONCLUSIONS

From the results we achieved so far we conclude that the proposed approach is promising for matching images based on their underlying sensor pattern noise. We will continue our investigations and aim to improve the method until it is robust enough to be deployed as a forensic tool.

ACKNOWLEDGEMENTS

This research has been funded with support from the European Commission under the 4NSEEK project with Grant Agreement 821966. This publication reflects the views only of the author, and the European Commission cannot be held responsible for any

use which may be made of the information contained therein.

REFERENCES

- Bayar, B. and Stamm, M. C. (2016). A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 5–10. ACM.
- Bayar, B. and Stamm, M. C. (2017). On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2152–2156. IEEE.
- Bayram, S., Sencar, H., Memon, N., and Avcibas, I. (2005). Source camera identification based on cfa interpolation. In *IEEE International Conference on Image Processing 2005*, volume 3, pages III–69. IEEE.
- Bondi, L., Baroffio, L., Güera, D., Bestagini, P., Delp, E. J., and Tubaro, S. (2016). First steps toward camera model identification with convolutional neural networks. *IEEE Signal Processing Letters*, 24(3):259–263.
- Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.
- Chen, C. and Stamm, M. C. (2015). Camera model identification framework using an ensemble of demosaicing features. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE.
- Chen, J., Kang, X., Liu, Y., and Wang, Z. J. (2015). Median filtering forensics based on convolutional neural networks. *IEEE Signal Processing Letters*, 22(11):1849–1853.
- Cozzolino, D. and Verdoliva, L. (2019). Noiseprint: a cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*.
- Geradts, Z. J., Bijhold, J., Kieft, M., Kurosawa, K., Kuroki, K., and Saitoh, N. (2001). Methods for identification of images acquired with digital cameras. In *Enabling technologies for law enforcement and security*, volume 4232, pages 505–513. International Society for Optics and Photonics.
- Gloe, T. and Böhme, R. (2010). The dresden image database for benchmarking digital image forensics. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1584–1590. Acm.
- Holst, G. (1998). *CCD arrays, cameras, and displays*. JCD Publishing.
- Kharrazi, M., Sencar, H. T., and Memon, N. (2004). Blind source camera identification. In *2004 International Conference on Image Processing, 2004. ICIP'04.*, volume 1, pages 709–712. IEEE.
- Kurosawa, K., Kuroki, K., and Saitoh, N. (1999). Ccd fingerprint method-identification of a video camera from videotaped images. In *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, volume 3, pages 537–540. IEEE.
- Li, C.-T. (2010). Source camera identification using enhanced sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 5(2):280–287.
- Lukáš, J., Fridrich, J., and Goljan, M. (2006). Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214.
- Mayer, O. and Stamm, M. C. (2018). Learned forensic source similarity for unknown camera models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2012–2016. IEEE.
- Mayer, O. and Stamm, M. C. (2019). Forensic similarity for digital images. *arXiv preprint arXiv:1902.04684*.
- Qian, Y., Dong, J., Wang, W., and Tan, T. (2015). Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, page 94090J. International Society for Optics and Photonics.
- Swaminathan, A., Wu, M., and Liu, K. R. (2007). Nonintrusive component forensics of visual sensors using output images. *IEEE Transactions on Information Forensics and Security*, 2(1):91–106.
- Tuama, A., Comby, F., and Chaumont, M. (2016). Camera model identification with the use of deep convolutional neural networks. In *2016 IEEE International workshop on information forensics and security (WIFS)*, pages 1–6. IEEE.
- Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155.
- Zhu, X., Qian, Y., Zhao, X., Sun, B., and Sun, Y. (2018). A deep learning approach to patch-based image inpainting forensics. *Signal Processing: Image Communication*, 67:90–99.